IxD Theory 2: Telecomunicazioni

IUAV University of Venice

Visual and Multimedia Communication graduate programme

Digitization and representation

Key concepts

1 Digital

2 Code

DIGITAL

To do with discrete amounts (from digit, digitus—counting on fingers—discrete counting).

Contrast with analog (a continuous spectrum).

Computers work by controlling the **flow** of electricity.

Computers use electrical voltages to represent binary numbers (base 2 numbers)

no voltage = 0   voltage = 1

CODE

So how do we get from an adding machine to the computers we use today?

CODE!

CODE

By code, I don't mean programming code but the more generic concept: making one thing stand for another.

E.g: Morse code, semaphore code, secret codes, etc.

Let's take Morse code. . .

Can anyone here do Morse code?

Morse code

A	.-
B	-...
C	-.-.
D	-..
E	.
F	..-.
G	--.
H	....
I	..
J	.---
K	-.-
L	.-..
M	--

N	-.
O	---
P	.--.
Q	--.-
R	.-.
S	...
T	-
U	..-
V	...-
W	.--
X	-..-
Y	-.--
Z	--..

Morse code uses combinations of 4 dots and dashes to **represent** letters of the alphabet.

When sent by radio the dots and dashes are **represented** by two different sounds.

Representation

Codes use one thing to represent another.

Computers can only manipulate binary numbers. We saw how by the clever combination of logical gates we can make the binary numbers represent decimal numbers.

But what if we want to manipulate things that are not numbers?

Representation

Anything else we want to manipulate must be **encoded** so it can be represented as numbers.

Representation

What might we want to represent in the computer?

Representation

Think of the programs we use:

- Photoshop
- Illustrator
- Flash animation
- Word processors
- Music programs
- 3D programs
- Databases
- Navigators.

Spreadsheets

Representation

What basic elements do these programs manipulate?

Representation

Photoshop – images: colour, position.

Representation

Illustrator – images: ?

Representation

Illustrator – images:

colour, position, lines, shapes and fills.

Representation

Flash – animation: ?

Representation

Flash – animation:

image, time, position.

Representation

Photoshop – images: colour, position

Illustrator – images:
colour, position, lines, shapes, fills

Flash – animation: image, time, position

Word – text: characters, words, paragraphs, position

Music programs – sound:
amplitude, frequency, time

3D programs – objects and scenes:
colour, light, 3D position, perspective

Spreadsheets - calculations: quantities,
mathematical operations, relationships

Databases – information and relationships:
objects, attributes, relationships.

Images

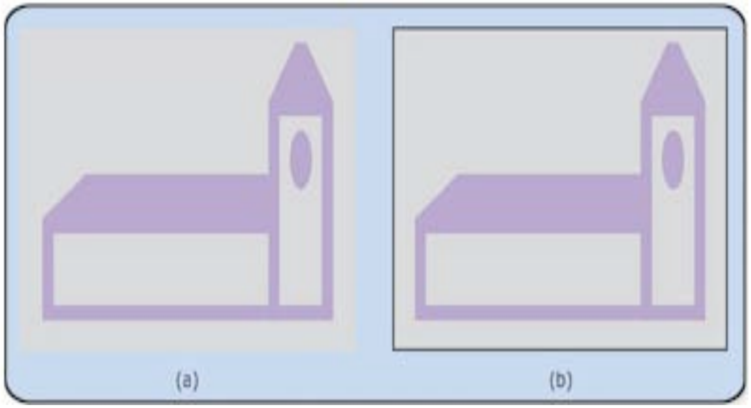
Images are stored as **pixels**, tiny dots of light on the screen.

They are stored in the computer in a special **video memory.**

Each location on the screen is referenced by **x and y coordinates**. Stored in each location is a binary number that tells the video controller what colour to display the pixel.

Images

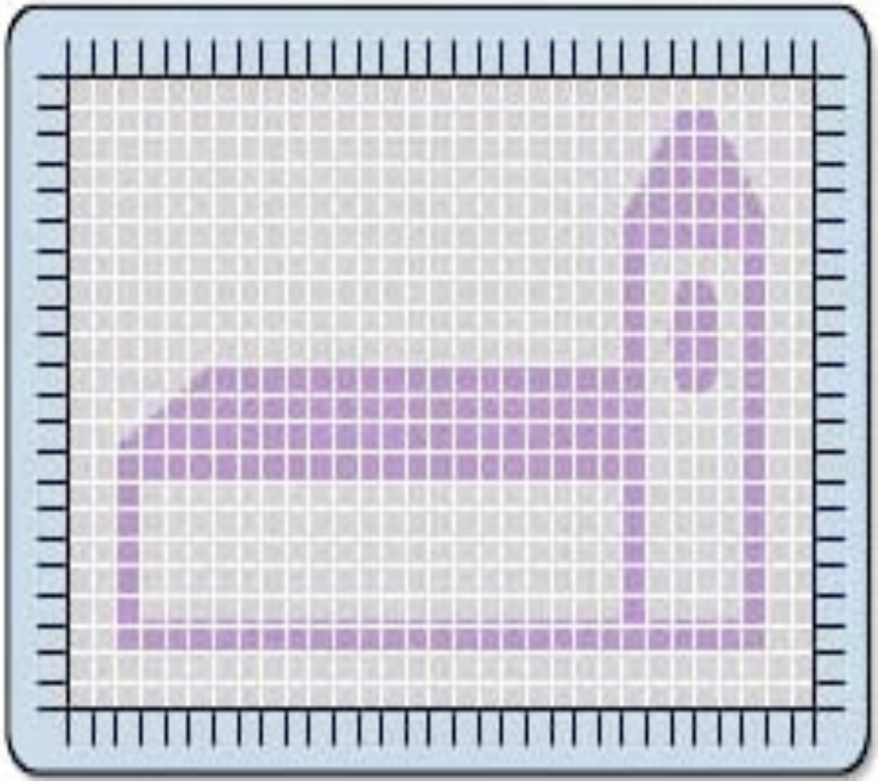
What does the computer do to display an image?



Define the boundaries of the image (b)

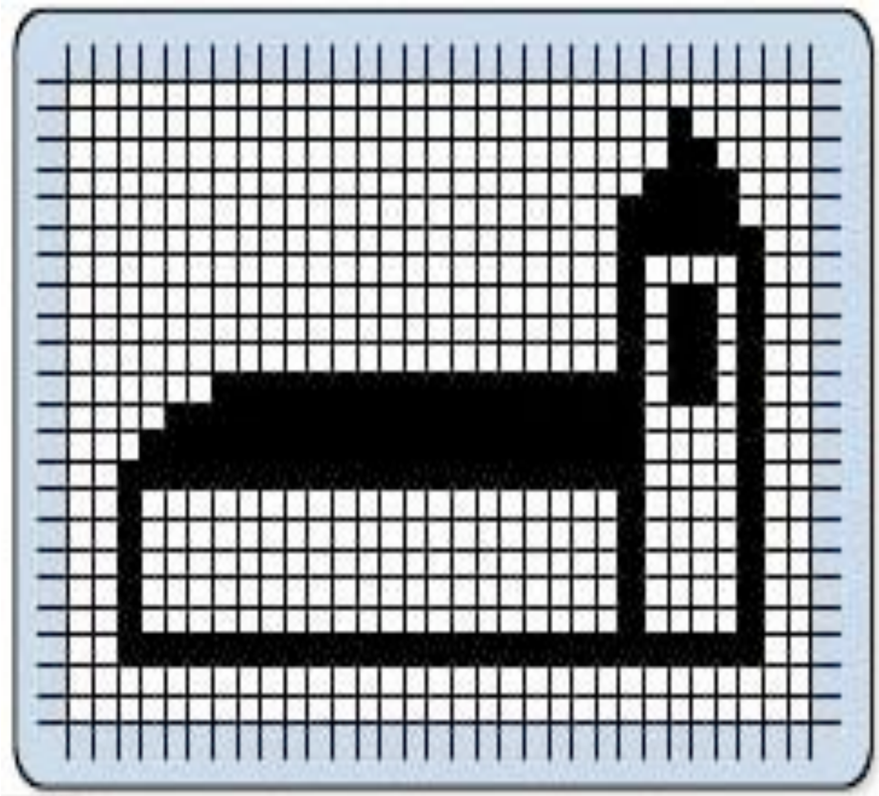
Images from Open University http://openlearn.open.ac.uk/mod/resource/view.php?id=32584

Images



The computer divides the image into squares: each square corresponds to one pixel on the screen.

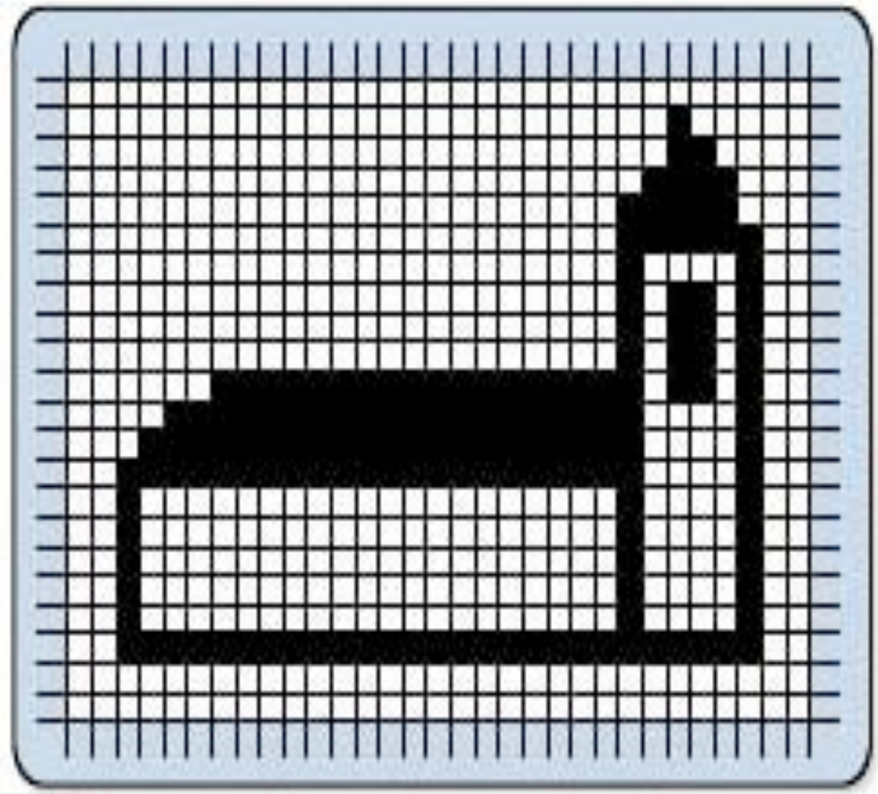
Images



If a square is more than 50% coloured, the pixel is given a value of **1.** If not, **0**.

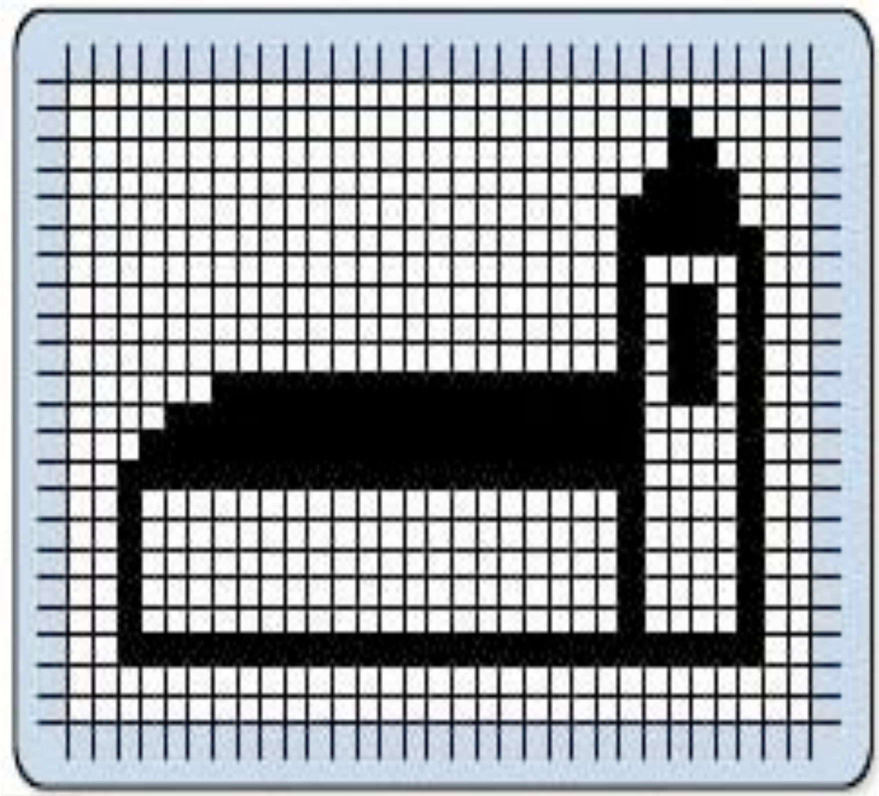
These values are stored in the video memory as a sequence of **0**s and **1**s.

Images



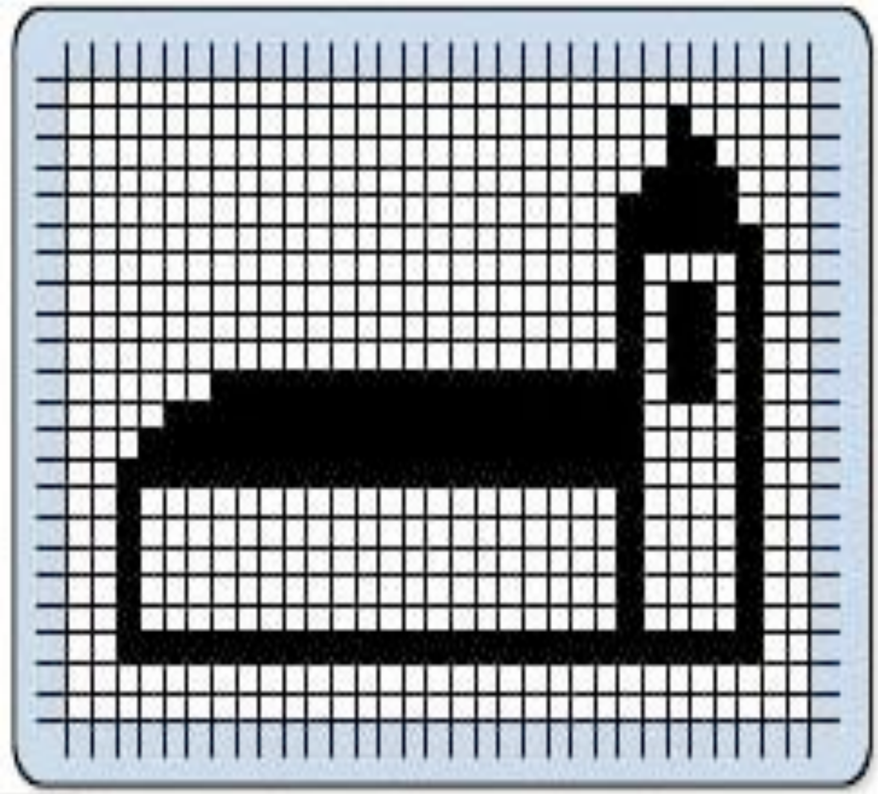
00000000000000000000000000000000

Images



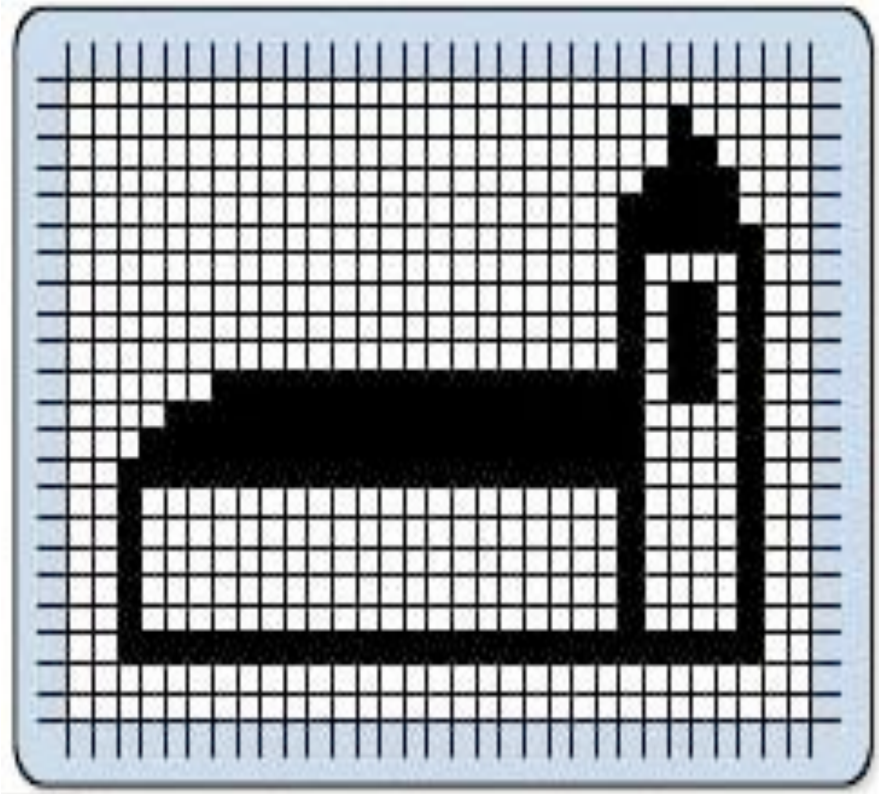
00000000000000000000000000000000
00000000000000000000000000100000

Images



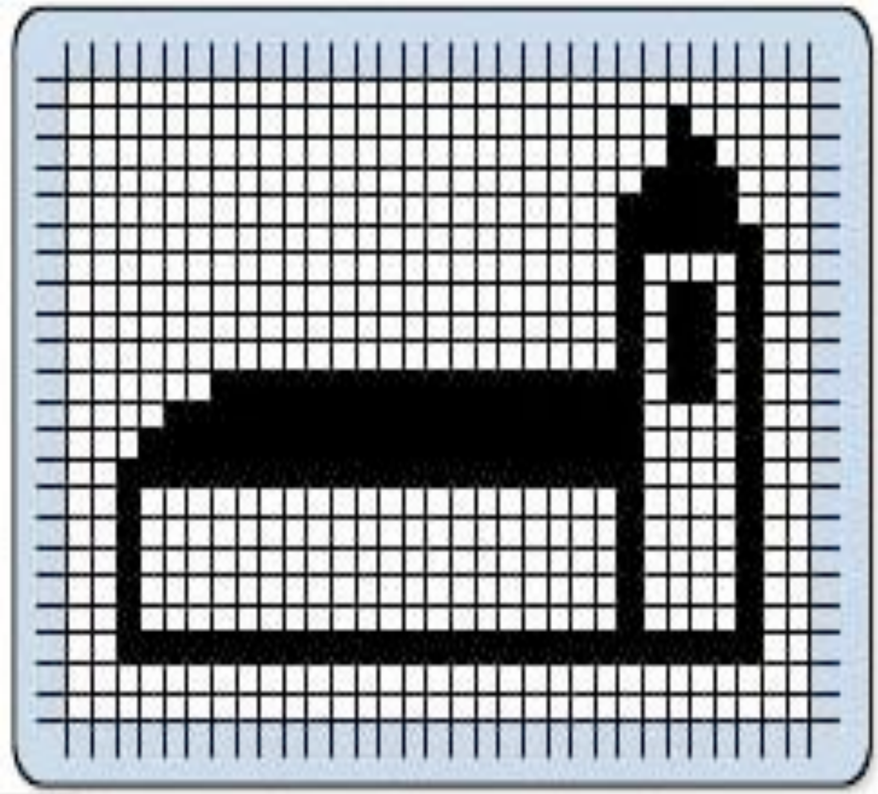
00000000000000000000000000000000
00000000000000000000000000100000
00000000000000000000000000110000

Images



00000000000000000000000000000000
00000000000000000000000000100000
00000000000000000000000000110000
00000000000000000000000001111000

Images



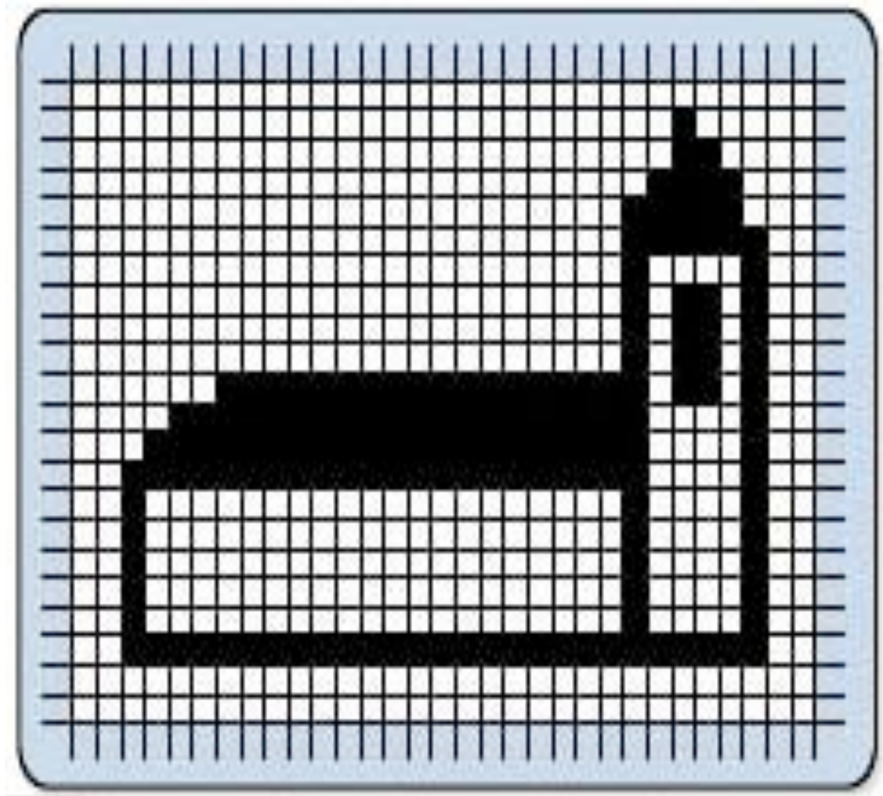
```
00000000000000000000000000000000
00000000000000000000000000100000
00000000000000000000000000110000
00000000000000000000000001111000
00000000000000000000000011111000
00000000000000000000000011111100
00000000000000000000000010000100
00000000000000000000000010110100
00000000000000000000000010110100
00000000000000000000000010110100
00000011111111111111111110110100
00001111111111111111111110000100
00011111111111111111111110000100
00111111111111111111111110000100
00100000000000000000000010000100
00100000000000000000000010000100
00100000000000000000000010000100
00100000000000000000000010000100
00100000000000000000000010000100
00111111111111111111111111111100
00000000000000000000000000000000
00000000000000000000000000000000
```

Pixels

If we only have a single binary number to represent each pixel, we can only have a monochrome picture.

In our example **0** = white, **1** = black.

But humans (although not horses) see in colour.



Coloured pixels

To represent colour, we need more than single binary numbers.

If we use a **byte** (8 bits) to represent the colour of a pixel. How many different colours would this give us?

Coloured pixels: 8 bit colour

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**

128 + 64 +32 +16 +8 +4 +2 +1 = 255

A byte can represent numbers between

0 and 255 i.e. 256 different colours.

Coloured pixels: simple 8-bit colour

1 byte = 8 bits = **256** different colours.

8-bit RGB

Red	3 bits = 4x2x1	= 8
Green	3 bits = 4x2x1	= 8
Blue	2 bits = 2x1	= 4

possible levels (the eye is less sensitive to blue).

Coloured pixels: 24-bit colour

256 is not very many colours, especially for designers.

If, instead of 1 byte per pixel, we use 3 bytes (8 bits for levels of red, 8 bits for levels of green, 8 bits for levels of blue), we get more gradations than the eye can distinguish.

Coloured pixels: 24-bit colour

1 byte each for

red, green, blue

256 x 256 x 256 gives

16 777 216 different colours.

Red = 255,0,0

Green = 0,255,0

Blue = 0,0,255

White = 255,255,255

Black = 0,0,0

Coloured pixels: look-up tables

A 1024 x 768 screen:

1 byte per pixel needs
786 432 bytes of memory;

3 bytes per pixel needs
2 359 296 bytes of memory
—over 2 megabytes.

Coloured pixels: look-up tables

Sometimes we don't have this amount of memory, or our computer is too slow to manipulate it quickly enough.

Coloured pixels: look-up tables

The look-up table method still has only 1 byte per pixel of video memory but instead of the number in memory representing the pixel colour directly, the memory contains a ‘pointer’ to a list of 256 x 24-bit colour numbers.

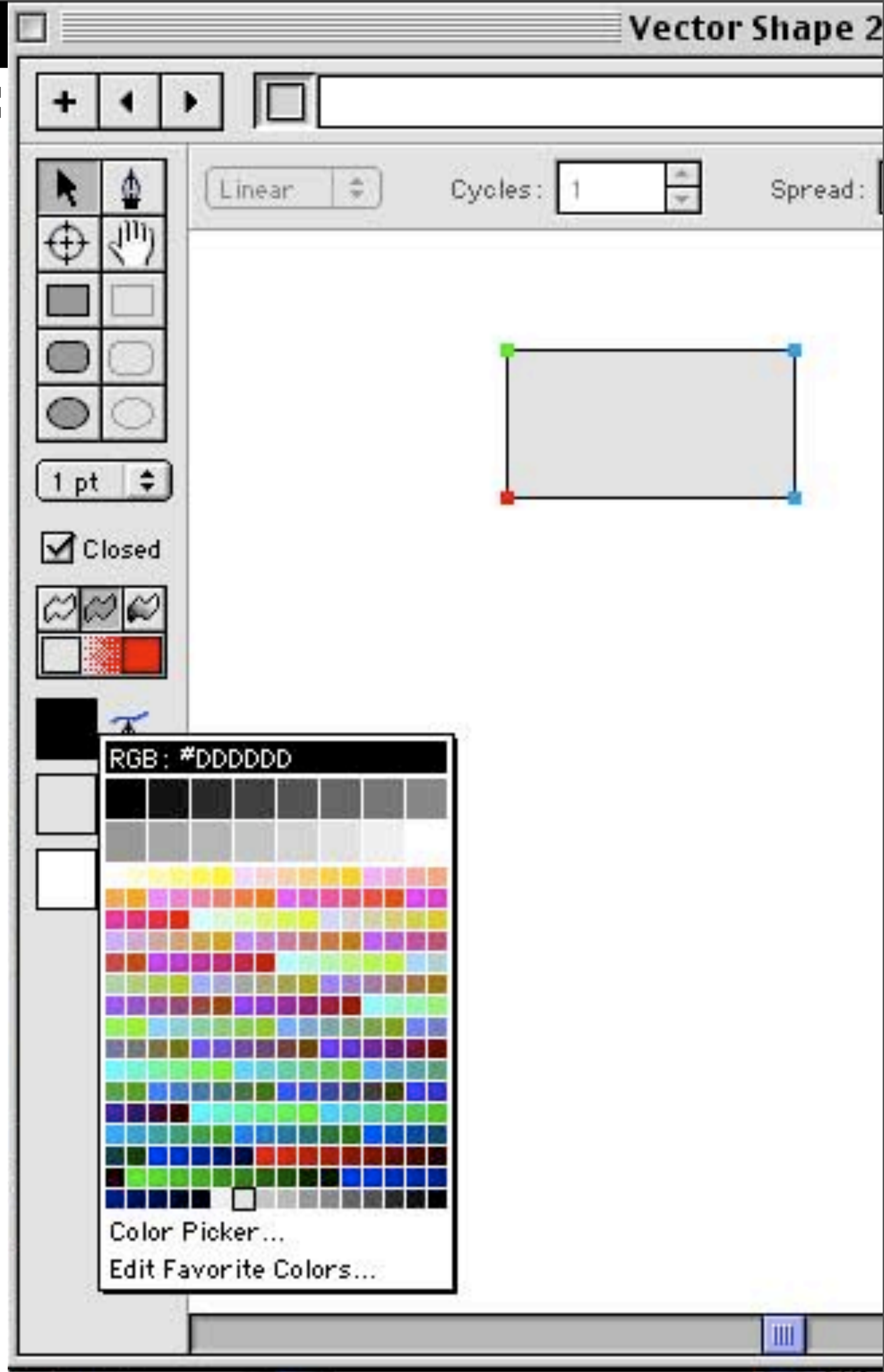
Coloured pixels: look-up tables

This means that colours can be any of the 16 million different colours available BUT you can only have 256 different colours on screen at once.

Coloured pixels: 8-bit look-up

	R	G	B
1	15	222	17
2	0	0	0
3	255	255	255
4	33	123	77
5	etc		
to			
256			

Coloured pixels: look-up tables



This is an example from Macromind Director: if we want special colours which don't appear in the normal palette, we can choose them from the full 16 million using the colour picker.

Summary: representation 1

- The computer can only manipulate numbers
- Things in the world which we want to manipulate by computer must be translated into numbers
- Images are digitized by the computer into an **array** of **pixels**
- Information about the colour of pixels is stored in **video memory**
- The computer's video controller scans the video memory and signals to the screen to display the correct colour for each pixel